**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Patent Application

for

# METHOD AND SYSTEM FOR MULTIPLICATION OF BINARY NUMBERS

Invention of: Clemens M. Zierhofer
Huettstrasse 50
A6250 Kundl
AUSTRIA

Attorney docket number: 1941/164

Attorneys:
Bromberg & Sunstein LLP
125 Summer Street
Boston, MA 02110-1618
Tel: (617) 443-9292
Fax: (617) 443-0004

Attorney Docket: 1941/164

# Method and System for Multiplication of Binary Numbers

## Cross Reference to Related Applications

This application claims priority from United States provisional application serial number 60/405,241, filed August 22, 2002, entitled "Method and System for Multiplication of Binary Numbers", the disclosure of which is incorporated herein by reference.

## Technical Field

The present invention relates to an efficient method and system for multiplying binary numbers.

## Background Art

Two N-bit binary numbers $A = [a_{N-1}\ a_{N-2} \ldots a_1\ a_0]$ and $B = [b_{N-1}\ b_{N-2} \ldots b_1\ b_0]$ are commonly multiplied as shown in Fig.1 (prior art). Here, the multiplication of two 8-bit numbers $A = 185_{dec} = [1\ 0\ 1\ 1\ 1\ 0\ 0\ 1]$ and $B = 237_{dec} = [1\ 1\ 1\ 0\ 1\ 1\ 0\ 1]$ to form a product 104 is depicted.

The product A*B is the sum of single partial products at particular binary positions. The single partial products are either number B or zero, dependent on the associated bit within number A. For example, the LSB of A, $a_0 = 1$, and thus the partial product at binary position $2^0$ is number $B = [1\ 1\ 1\ 0\ 1\ 1\ 0\ 1]$. The neighboring bit $a_1 = 0$, and thus the associated partial product is $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$. As used in this description and the accompanying claims, the array of partial products 102 formed by this common method of multiplication shall be referred to as a "diamond-array". The diamond-array 102 depicted in Fig. 1 requires that 64 digits be added to form the product.

Fig. 2 (prior art) shows multiplication using a well-known Booth scheme. Multiplication using the Booth methodology reduces the number of partial products in a diamond-array by a factor of 2. This is accomplished by representing one of the multiplicands by numbers whose binary weights differ by at least a factor 4. However, the Booth scheme disadvantageously requires that negative numbers be introduced. For example, number $A = 185_{dec} = [1\ 0\ 1\ 1\ 1\ 0\ 0\ 1]$ is commonly regarded as a sum of

positive terms: $2^7+2^5+2^4+2^3+2^0$. Following the Booth scheme, the number A is instead represented as $2^8-2^6-2*2^2+2^0$. Thus, the negative partial sums have to be represented as twos compliment numbers.

Note that at least every second binary position of A is necessarily zero. However, the representation of negative numbers by twos compliment requires leading sequences of ones, which significantly reduce the benefit of the approximately 50% reduction in the number of partial products 201. For N = 8, 60 digits remain to be added using the Booth methodology.

## Summary

In accordance with one aspect of the invention, a multiplier for multiplying a first binary number $a = [a_{N-1} \ldots a_1 a_0]$ and a second binary number $b = [b_{N-1} \ldots b_1 b_0]$ is presented. The multiplier includes a first port for receiving a first signal representing the binary number a, and a second port for receiving a second signal representing the binary number b. A triangle array is operatively coupled to the first signal and the second signal. An adder adds elements of the triangle array to produce a third signal representing a product of the first signal and the second signal.

In accordance with another aspect of the invention, a processor for multiplying a first signal representing a first binary number $a = [a_{N-1} \ldots a_1 a_0]$ and a second signal representing a second binary number $b = [b_{N-1} \ldots b_1 b_0]$ is presented. The processor includes input means for receiving the first signal and the second signal. The processor also includes means for forming a triangle array as a function of the first signal and the second signal. An adder adds elements of the triangle array to form a third signal representing a product of the first signal and the second signal.

In accordance with still another aspect of the invention, a method for performing digital signal processing that requires multiplication of a first signal representing a first binary number $a = [a_{N-1} \ldots a_1 a_0]$ and a second signal representing a second binary number $b = [b_{N-1} \ldots b_1 b_0]$ is presented. The method includes receiving the first signal and the second signal. A triangle array is formed from the first binary number and the second binary number, the triangle array including lines k = 0 to N-1. The lines k = 0 to N-1 are added to produce a third signal representing a product of the first signal and the second signal.

In accordance with yet another aspect of the invention, a computer program product for use on a computer system for multiplying a first binary number $a = [a_{N-1} \ldots a_1 a_0]$ and a second binary number $b = [b_{N-1} \ldots b_1 b_0]$ is presented. The computer program product includes a computer usable medium having computer readable program code thereon. The computer readable program code includes program code for forming a triangle array from the first binary number and the second binary number, the triangle array including lines k = 0 to N-1. The computer readable program code also includes program code for adding lines k = 0 to N-1.

In accordance with embodiments related to the above-described embodiments of the invention, the triangle array includes lines k = 0 to N-1, such that line k = 0 of the triangle array is equal to $[0\ (a_0 * b_0)]$; and the lines k = 1 to N-1 of the triangle array are equal to $[p_1\ p0\ d_{k-1} \ldots d_1\ d_0]$. The peak bits $[p_1\ p0]$ for each line k are determined by $[p_1\ p0] = [0\ 0]$ if $a_k * b_k \neq 1$; $[p1\ p0] = [1\ 0]$ if $[a_k * b_k = 1$ AND $c_k = 1]$; and $[p1\ p0] = [0\ 1]$ if $[a_k * b_k = 1$ AND $c_k = 0]$. The bits $[d_{k-1} \ldots d_1\ d_0]$ for each line k are determined by: $[d_{k-1} \ldots d_1\ d_0] = [0_{k-1} \ldots 0_0]$ if $[a_k\ b_k] = [0\ 0]$; $[d_{k-1} \ldots d_1\ d_0] = [a_{k-1} \ldots a_1\ a_0]$ if $[a_k\ b_k] = [0\ 1]$; $[d_{k-1} \ldots d_1\ d_0] = [b_{k-1} \ldots b_1\ b_0]$ if $[a_k\ b_k] = [1\ 0]$; and $[d_{k-1} \ldots d_1\ d_0] = [s_{k-1} \ldots s_1\ s_0]$ if $[a_k\ b_k] = [1\ 1]$. The sequence $s = [s_{N-2} \ldots s_1\ s_0]$ is equal to the sum sequence $[(a_{N-2} + b_{N-2}) \ldots (a_1+b_1)\ (a_0+b_0)]$ and $c = [c_{N-1} \ldots c_1]$ is equal to the carry sequence associated with the sum sequence s. The triangle array thus formed may be represented by a number of digits that is approximately 20-50% less than the number of digits required in a diamond array.

## Brief Description of the Drawings

The foregoing features of the invention will be more readily understood by reference to the following detailed description, taken with reference to the accompanying drawings, in which:

Fig. 1 depicts prior art multiplication of two N-bit binary numbers A and B;

Fig. 2 depicts prior art multiplication of the binary numbers A and B employing Booth's algorithm;

Fig. 3 depicts diamond arrays with rotated V's, in accordance with one embodiment of the invention;

Fig. 4 is a table that shows possible peak-bit and branch configurations of a V, in accordance with one embodiment of the invention;

Fig. 5 depicts the addition of two branch numbers without the most significant bit, resulting in a sum sequence and a carry sequence, in accordance with one embodiment of the invention;

Fig. 6 depicts lower branch values when multiplying the numbers A and B, in accordance with one embodiment of the invention;

Fig. 7 depicts a resulting multiplication scheme prior to removal of zero bits positioned above pairs of peak-bits, in accordance with one embodiment of the invention;

Fig. 8 depicts a resulting multiplication scheme with zero bits positioned above the pairs of peak-bits removed, in accordance with one embodiment of the invention;

Fig. 9 is a table depicting the switching of bit sequences based on bits $a_k$ and $b_k$, in accordance with one embodiment of the invention;

Fig. 10 is a block diagram of a multiplier for multiplying a first binary number and a second binary number, in accordance with one embodiment of the invention.

## Description

A method and system for efficiently multiplying binary numbers is presented. In particular, the method and system includes reducing the number of digits used in connection with partial products formed during multiplication. Details of various embodiments are discussed below.

Fig. 3 shows an advantageous way of looking at the diamond-array 102 of Fig.1, in accordance with one embodiment of the invention. Instead of looking at the lines of the array, the array can be regarded as composed of structures similar to "rotated V's", whose peaks are looking to the left-lower corner. The V's have the following general properties:

(1) Each V has one peak-bit 301 and two branches 303 and 304 with an equal number of bits, respectively. The number of bits within a branch can also be zero. For example, a V may consists of only the peak-bit.

(2) The peak-bit 301 of each V is the product of the bits of equal binary position within numbers A and B, respectively. In Fig.3, two V's 301 are highlighted. The corresponding peak-bits (bold) 301 are the products of the first bits (MSB) $a_7b_7 = 1$, and bits $a_3b_3 = 0$, respectively (see Fig. 4, discussed below).

(3) The branches 303 and 304 include either truncated versions of numbers A or B, or zeros only. In Fig.3, the V 301 with peak-bit "1" has an upper branch containing 7 bits of A (i.e., $[a_6 \ldots a_1\, a_0]= [0\ 1\ 0\ 1\ 0\ 0\ 1]$), and a lower branch consisting of 7 bits

of B (i.e., $[b_6 \ldots b_1 b_0] = [1\ 1\ 0\ 1\ 1\ 0\ 1]$). The V 301 with peak bit "0" shows an upper branch 303 composed of 4 zeros (i.e., [0 0 0 0] ) and a lower branch 304 composed of 4 bits of A (i.e., $[b_3 b_2 b_1 b_0] = [1\ 1\ 0\ 1]$).

(4) Four possible configurations can occur, determined by the two bits within A and B, that can be used to determine the peak-bit. Fig. 4 is a table that shows these four configurations, in accordance with one embodiment of the invention.

(5) The overall diamond-array is fully covered by exactly N non-overlapping V's.

The V's are defined by bits $a_k$ and $b_k$ at binary position k in numbers A and B, as summarized in Fig. 4. If both bits $a_k$ and $b_k$ are zero, both branches contain only zeros. For $a_k = 0$ and $b_k = 1$, bit sequence $[a_{k-1} \ldots a_1 a_0]$ appears in the upper branch, and for $a_k = 1$ and $b_k = 0$, sequence $[b_{k-1} \ldots b_1 b_0]$ appears in the lower branch. For $a_k = 1$ and $b_k = 1$, both sequences $[a_{k-1} \ldots a_1 a_0]$ and $[b_{k-1} \ldots b_1 b_0]$ have to be considered. In general, bits $a_k$ and $b_k$ can be regarded as switches, where bits $a_k$ activate or deactivate the truncated versions of number B, and bits $b_k$ activate or deactivate the truncated versions of number A.

Using the commutative law, upper and lower branches of the V's can be flipped arbitrarily, without changing the overall sum. For example, all zero-branches can be flipped such that they become upper branches. This causes a concentration of zeros in the upper left area of the diamond-array, that is, the region above the line of peak-bit elements. Only branches of V's whose peak-bit is "1" may contain non-zero elements in this region. To also remove these, the two branches are added, and the sum positioned in the lower branches. This addition does not need to be done individually for each V of particular length. Instead, it can be done once by adding sequences $[a_{N-2} \ldots a_1 a_0]$ and $[b_{N-2} \ldots b_1 b_0]$, for example, numbers A and B without the most significant bit, respectively. The results are sum-sequence $[s_{N-2} \ldots s_1 s_0]$, and the carry-sequence $[c_{N-1} \ldots c_2 c_1]$. For the present example, these sequences are shown in Fig.5.

The carries at each bit position are of interest, since different V's have different lengths of branches. If the overall carry after addition of the two branches is "1", the binary position of this carry is equal to the binary position of the peak-bit "1". Addition of the two "1's" results in "10". This leads to the following simple rule: if a carry appears, the peak-bit "1" is shifted one position to the left. In the following, single peak-bits are replaced by pairs of peak-bits $[p_1 p_0]$. The resulting "lower branches" 602 are

summarized for all branch-lengths in Fig. 6, in accordance with one embodiment of the invention.

Note that the first two bits in each line of lower branches 602 in Fig. 6 represent the pair of peak-bits [p1 p0] (bold), which are either "01" (no carry) or "10" (with carry). The remaining bits are bit sequences from the sum in Fig. 5, ranging from lengths 1 to 7.

After flipping branches of V's with peak-bit "0" as described above, and representing V's with peak-bit "1" as lower branches 602 according to Fig.6, the multiplication scheme looks like Fig.7, in accordance with one embodiment of the invention. Note that the single peak-bits are replaced by a pair of bits (bold).

Now all bits above the pairs of peak-bits are zero and thus can be omitted as shown in Fig.8, in accordance with one embodiment of the invention. As used in this description and the accompanying claims, the resulting array shall generally be referred to as a "triangle-array."

As compared to the scheme depicted in Fig.1, the diamond-array 102 with exactly $N^2 = 64$ digits has changed to the triangle-array 801 with exactly $\frac{N(N+3)}{2} = 44$ digits, a bit-reduction of approximately 30%. For larger N, a bit-reduction of roughly 50% may be achieved. This reduction also compares favorably to the Booth-algorithm, which requires 60 digits to be added for N = 8, considerably more than the 44 digits required for N = 8 in the present invention.

Fig. 10 shows a block diagram of a multiplier 1000 for multiplying a first signal and a second signal, in accordance with one embodiment of the invention. The first signal represents a first binary number $a = [a_{N-1} \ldots a_1 a_0]$, and the second signal represents a second binary number $b = [b_{N-1} \ldots b_1 b_0]$. Multiplier may be, without limitation, a device such as a communications device, a signal processor, a microprocessor, central processor, and/or computer for operating on data signals.

The multiplier includes a first port 1001 for receiving the first signal 1005. A second port 1002 receives the second signal 1006.

Operatively coupled to the first port 1001 and the second port 1002 is a triangle-array 1003. The triangle array 1003 may be represented in, without limitation, a tangible medium, such as a register and/or computer readable medium. For example, the triangle-array 1003 may be stored, without limitation, on a diskette, CD-ROM, ROM, RAM, or fixed disk.

The triangle array 1003 includes lines k = 0 to N-1. Line k = 0 of the triangle array is equal to [0 ($a_0 * b_0$)]. Line k = 0 equal to [$p_1$ $p_0$] = [1 1] cannot occur.

Lines k = 1 to N-1 of the triangle array are equal to [$p_1$ p0 $d_{K-1}$ ... $d_1$ $d_0$]. The pair of peak-bits [$p_1$ $p_0$] in each line k = 1 to N-1 is different from [0 0], only if $a_k * b_N = 1$. When $a_k * b_k = 1$, [$p_1$ $p_0$] = [0 1] for $c_k = 0$, and [$p_1$ $p_0$] = [1 0] for $c_k = 1$.

Each digit in the triangle-array below the pairs of peak-bits (i.e., [$d_{k-1}$ ... $d_1$ $d_0$]) may be controlled by bits $a_k$ and $b_k$ according to the table shown in Fig. 9, in accordance with one embodiment of the invention. In particular, for each line k, [$d_{k-1}$ ... $d_1$ $d_0$] is set to: all zeros, if [$a_k$ $b_k$] = [0 0]; [$a_{k-1}$ ... $a_1$ $a_0$] if [$a_k$ $b_k$] = [0 1]; [$b_{k-1}$ ... $b_1$ $b_0$] if [$a_k$ $b_k$] = [1 0]; or [$s_{k-1}$ ... $s_1$ $s_0$] if [$a_k$ $b_k$] = [1 1]. Such control may be implemented using a multiplexer for each digit, which may be a 4 to 1 multiplexer. The inputs into the multiplexer can be [0 0 . . . 0], [$a_{k-1}$ ... $a_1$ $a_0$], [$b_{k-1}$ ... $b_1$ $b_0$], or [$s_{k-2}$ ... $s_1$ $s_0$], which are controlled by $a_k$ and $b_k$.

An adder 1004 adds the lines k in the triangle-array 1003 to produce a third signal 1007. The third signal 1007 represents a product of the first signal 1005 and the second signal 1006.

In accordance with various embodiments of the invention, the multiplier may include an additional (k-1)-bit adder for addition of numbers A and B without most significant bit's to compute the sum sequence [$s_{N-2}$ ... $s_1$ $s_0$], and the carry-sequence [$c_{N-1}$ ... $c_2$ $c_1$]. See Fig. 5 and accompanying text for details regarding the sum and carry sequence.

Squaring of binary numbers represents a special case of the method and system described herein. In a squaring scheme, only symmetrical V's occur, that is, they are composed of either 2 branches of zeros, or 2 branches each containing a truncated version of the number to be squared. The adder to compute the sum-sequence [$s_{N-2}$ ... $s_1$ $s_0$] and the carry-sequence [$c_{N-1}$ ... $c_2$ $c_1$] can be omitted, since the addition of two equal numbers can trivially be achieved by shifting the number to the left and adding a tailing zero. Additionally, the 4-to-1 multiplexers can be replaced by simple AND-gates.

In various embodiments of the invention, the disclosed system and method for multiplying binary numbers may be implemented as a computer program product for use with a computer system or processor. Such implementation may include a series of computer instructions fixed either on a tangible medium, such as a computer readable medium (e.g., a diskette, CD-ROM, ROM, or fixed disk) or transmittable to a computer system, via a modem or other interface device, such as a communications adaptor

connected to a network over a medium. The medium may be either a tangible (e.g., optical or analog communications lines) or a medium implemented with wireless techniques (e.g., microwave, infrared, or other transmission techniques). The series of computer instructions embodies all or part of the functionality previously described herein with respect to the system and method. Those skilled in the art should appreciate that such computer instructions can be written in a number of programming languages for use with may computer architectures or operating systems. Further, such instructions may be stored in any memory device, such as a semiconductor, magnetic, optical or other memory devices, and may be transmitted using any communications technology, such as optical, infrared, microwave, or other transmission technologies. It is expected that such a computer program product may be distributed as a removable medium with accompanying printed or electronic documentation (e.g., shrink wrapped software), pre-loaded with a computer system (e.g., on system ROM or fixed disk), or distributed from a server or electronic bulletin board over a network (e.g., the Internet or World Wide Web). Of course, some embodiments of the invention may be implemented as a combination of both software (e.g., a computer program product) and hardware. Still other embodiments of the invention are implemented as entirely hardware, as discussed previously, or entirely software (e.g., a computer program product).

The present invention may be embodied in still other specific forms without departing from the true scope of the invention. The described embodiments are to be considered in all respects only as illustrative and not restrictive.